

```
A>BUBBLE.CML                                     ProgFree: 6.1296
0040 // Modulet basis indeholdende en ombytningsprocedure
0050 //
0060 MODULE basis
0070   PROC ombyt(REF a,REF b)
0080     temp:=a
0090     a:=b
0100     b:=temp
0110   ENDPROC ombyt
0120 ENDMODULE basis
0130 //
0140 // Modulet bubble indeholdende sorteringsproceduren bubblesort
0150 //
0160 MODULE bubble
0170   USE basis
0180   PROC bubblesort(REF tabel(),maxindeks)
0190     TRAP
0200     IF maxindeks>20 THEN
0210       REPORT
0220     ENDIF
0230     i:=1
0240     ombytning:=TRUE
0250     WHILE (i<=maxindeks-1) AND (ombytning) DO
0260       j:=1
1LIST 2RUN 3AUTO 4RENUM 5DIR 6DIR CML7CON 8LOAD 9FIND 0SEL CON
```

UniComal

ET DANSK

PROGRAMMERINGSSPROG

Af Henrik Nellager

■ Comal er et programmeringssprog, der for alvor blev lanceret i starten af det årti, som nu rinder ud. Det var blandt andre danskeren Børge Christensen, der var med til at definere sproget, der er en blanding af Basic og Pascal.

Kort fortalt ses ligheden med Basic gennem brug af linjenumre, uerklærede simple variable og det faktum, at Comal oprindeligt er et interaktivt sprog, hvorfor traditionelle Comal-systemer leveres med en fortolker og ikke en oversætter.

Analogien til Pascal trækkes gennem opbygning af strukturerede størrelser som f.eks. procedurer og funktioner og dermed også introduktion af et begreb som rekursion.

Det er ikke ofte, at Comal-pakker til PC'er frigives, men nu er *UniComal version 3.00* en realitet. Det er det danske firma af samme navn, der står bag systemet, og der er sket en del siden Comal blev erklæret officielt sprog i folkeskolerne i starten af 80'erne.

Den gamle Comal-skål er blevet udbygget, men ikke blot selve sprogkernen er blevet ekspanderet; antallet og omfanget af medfølgende biblioteker er blevet betragteligt stort. Der er eksempelvis moduler til musestyring, grafik og styring af eksterne enheder gennem den serielle port, der følger med i UniComal-pakken.

Desuden indeholder UniComal version 3.00 ikke blot en Comal-fortolker, men også en Comal-oversætter. Det er derfor muligt at oversætte et syntaktisk fejlfrit program til en fra DOS eksekverbar fil, hvilket giver mulighed for at skjule kildeteksten. Dette er som bekendt ikke muligt i fortolkerbaserede systemer.

Efter denne lille forsmag er det vist tid til at åbne pakken.

Rod i manualerne

UniComal version 3.00 leveres i to æsker, og de indeholder i alt to ringbind og seks 5,25"-disketter foruden manualerne i løbblad-

form. Disse skal man selv pakke ud og sætte ind i ringbindene.

Der leveres først og fremmest en referencemanual på ca. 500 sider. Denne omhandler de enkelte kommandoer i det interaktive Comal-system. Dertil kommer der mindre manualer til de enkelte biblioteksmoduler og en mindre sag kaldet Comal and DOS, der forklarer om samspillet mellem UniComal og operativsystemet samt behandler standardopætningsproblematikken.

En manual til den medfølgende Comal-oversætter var endnu ikke færdig, da vi lånte programmet til anmeldelse.

For den uøvede bruger indeholder manualerne en hjælp til installation, men så er alt også sagt. Der er ingen forklaring til selve sproget Comal, og den interaktive dialog er kun beskrevet gennem de enkelte stikord i referencemanualen. Det betyder, at man har svært ved at få et overblik over editorens funktioner og virkemåde, og der er faktisk kun den mulighed, at man eksperimenterer sig frem.

Der er gengivet masser af programeksempler, men det kræver Comal-kendskab i større stil og desuden bred programmerings erfaring ved de større ting at kunne forstå årsag og sammenhæng.

Manualerne fremstår derfor som særdeles ekspertrelaterede, og en ukyndig Comal-programmør vil blive stillet overfor en del indledende gymnastik, før systemet fungerer, som det ønskes.

Endelig bør det pointeres, at manualerne er på engelsk. Dette hænger sammen med, at UniComal lancerer deres projekt internationalt.

Syntaksstyret editor

Den interaktive Comal-del kan startes op på to måder: Med eller uden brug af matematisk co-processor. Umiddelbart fra start er systemet klart til at modtage kommandoer fra brugeren.

Skærbilledet er særdeles pri-

mitivt. Der er intet givet, udover den allernederste linje, hvor det står angivet, hvordan de 10 funktionstaster er defineret. Der er således ingen vinduer, ej heller nogen form for menustyring. Det er en traditionel, interaktiv brugergrænseflade.

Systemet kan modtage kommandoer direkte. Blandt de mange muligheder kan nævnes kommandoer til at hente og gemme filer, liste det aktuelle program, køre det, kigge på harddisken, ændre filkatalog og foretage udskrivning.

En af de mange muligheder er så at påbegynde indtastningen af et program eller hente et fra disketten. Som led i COMAL-filosofien er der linjenumre i programmet, men disse har nu ingen betydning overhovedet. Gamle Basic-programmører vil kunne erindre, at man dér kan hoppe rundt mellem de enkelte linjer ved en GOTO-kommando efterfulgt af et linjenummer. I UniComal er alt den slags baseret på etiketter, hvorfor linjenumrenes funktion er fjernet.

Man kan så indsætte linjer på kryds og tværs imellem hinanden, og man kan også rette direkte på skærmen i en vilkårlig vist linje. Man stiller sig simpelt hen på den ønskede linje, foretager rettelsen og konfirmerer med et linjeskift. Editoren er således ikke linjebaseret, men skærmorienteret.

Ved rulning af skærmen er det også muligt på let vis at bladre frem og tilbage i programmer på mere end en skærmfuld linjer.

Det måske allerbedste ved denne editor er, at den er syntaksstyret. Den kender derfor COMAL-sprogets udseende, og hvis der er glemt nøgleord, bliver disse automatisk indsat. Ved definitioner af moduler, ved forgreninger og løkker – og andre lignende steder – foretages desuden automatisk indrykning af de indlejrede komponenter. Dette fremmer læsbarheden af et program, og programmøren slipper for selv at

tage hånd om dette. Til gengæld kan man så heller ikke selv bestemme stilen.

Opstår der syntaksfejl i en indtastet linje, meldes dette øjeblikkeligt som fejl, da det netop er en fortolker, vi har med at gøre. Her er fejlmeddelelsen ikke altid lige sigende, men det er vel et vanligt problem i produkter med programmeringssprog.

Alt i alt er editoren derfor typisk interaktiv, og f.eks. en Turbo Pascal-programmør vil få sine problemer med at omstille sig fra dens editor til den her givne. De er som nat og dag – bortset fra, at begge modtager tryk fra tastaturet og gengiver dette på skærmen.

Endelig bør det nævnes, at vi ledte forgæves efter et hjælpesystem. Der er intet on-line kontekst-sensitivt hjælpesystem, der kan klare dig ud af vanskelighederne. Der kan kun henvises til referencemanualen.

Lidt om sproget

Vi har allerede givet en forsmag på, hvordan Comal arbejder. Væsentligt er det dog at bemærke, at UniComal har udvidet den vanlige kerne af konstruktioner.

Det er nu blevet muligt at operere med pegervariable. Dette muliggør brugen af dynamiske datastrukturer, hvilket er slemt nødvendigt ved implementation af abstrakte datastrukturer som hængende lister, træer, køer, stakke m.v.

Desuden er typebegrebet blevet udvidet. Hidtil har der kun været mulighed for simple variable, hvilket vil sige tegn, heltal, reelle tal, logiske værdier og lignende, og sammensatte variable i tabeludgaver. Nu er det blevet muligt at definere en slags poster, kaldet strukturer, hvor hver enkelt struktur indeholder en række felter, der kan have forskellige typer. Dér er UniComal-folkene dog ikke stoppet, for strukturer krytter data sammen med deres operationer; procedurer og funktioner. Dette kendes måske bedst fra Turbo Pascals objektorienterede

Det danske firma UniComal har frigivet en ny version af deres fortolker til programmeringssproget Comal. Den indeholder en stærkt udvidet Comal-skål og en lang række standardbiblioteker, og tilmed leveres UniComal-systemet også med en Comal-oversætter.

programmeringsdel, der kalder disse sammensatte størrelser for objekter.

UniComal bruger derfor konceptet objektorienteret programmering som værende en del af deres filosofi bag systemet. Desværre er strukturer ikke helt så effektive som Turbo Pascals objekter, da der opereres med statisk binding af variable. Polymorfisme er derfor ikke muligt, og strukturer er derfor ikke specielt fleksible. De kan samle data og operationer i en logisk form, men deres betydning bør ikke overdrives.

Herudover er UniComal blevet forsynet med et modulbegreb. Det betyder, at de enkelte dele af et program kan opsplittes i moduler, der kan konstrueres og afprøves selvstændigt. Disse har så en grænseflade, som brugeren af modulet kan kigge på – selve implementationen er naturligvis skjult for brugeren. Det betyder også, at programmøren kan implementere sit modul, som det nu er passende, blot specifikationen i grænsefladen er overholdt.

En senere optimering som følge af en videreudvikling vil så ligge i modulernes implementationssektioner, og de større programmer forbliver principielt uforandrede deraf. Dette spiller en stor rolle ved større projekter, f.eks. af industriel beskaffenhed, hvor pålidelighed er en magtfaktor. En ændring af et kompliceret reguleringssystem vil ikke være nødvendigt blot på grund af mindre ændringer i de anvendte komponenters implementation.

UniComal introducerer desforuden en særdeles smidig form for fejlbehandling, hvilket kan ses illustreret på det lille program vist her på siderne. Så snart der opstår en fejl, kan programmøren tvinge programmet over i en fejlbehandlingsdel, og ved dens afslutning kan der returneres enten tilbage eller til andre dele.

Det er en subjektiv afgørelse, hvorvidt Comal-sproget findes behageligt at arbejde med. Er

man vant til at benytte andre programmeringssprog, f.eks. Pascal, kan det forekomme tungt og ikke mindst ustruktureret; til trods for de nytilkomne konstruktioner.

Grafik og musestyring

Som nævnt leveres UniComal med en række biblioteker, og vi skal kort beskrive disse et efter et.

I grafikmodulet ligger der alskens kommandoer til styring af en grafiskskærm. Der kan specificeres vinduer og farver, og en række kommandoer gør det muligt at tegne streger, cirkelbuer, cirkler, og polygoner. Desuden er der indbygget kommandoer til skildpaddeskrift i Logo-stil.

Grafikmodulet understøtter brug af grafikkortene CGA, EGA, MCGA og VGA. Alt i alt må det siges, at grafikmodulet er særdeles udviklet og indeholder alt, hvad man kan forlange.

Et andet hedder UniMouse, og med dette kan man indlægge musestyring i sine applikationer. Dertil kræves en mus, der kommunikerer gennem en standard Microsoft driver på interrupt 33h. Også PS/2-musen er understøttet. Ellers er modulet i grove træk blot en Comal-udgave af de kommandoer, som driveren giver. Det betyder, at det er muligt at kontrollere, om der er trykket på en tast, hvor musen så står, hvor hurtigt den skal rykke sig m.m. Modulet understøtter også brug af mus på en grafiskskærm.

UniMouse er meget nyttigt og anvendeligt ved konstruktion af applikationer, der benytter mus – og man slipper for at gå dybere ned i niveau på den måde, som to artikler i hhv. "Alt om Data" nr. 9/89 og 10/89 angav.

Seriell kommunikation

Et tredje modul indeholder systemrutiner. Herved forstås procedurer og funktioner, der foretager kommunikation mellem et Comal-program og dets omgivel-

ser.

Blandt de mange muligheder skal der her nævnes nogle få stykker. En funktion giver den streng, som SET-kommandoen fra DOS har specificeret, og en anden kontrollerer for, om der findes en matematisk co-processor i PC'en. En procedure lægger direkte en oktet ned på en plads i adresseringsområdet, og en anden sætter farven på kanten af en farvegrafikskærm.

Som det forhåbentlig fremgår af dette sporadiske kig på mulighederne, indeholder systemmodulet en række vidt forskellige funktioner, der tilsammen kan gøre livet meget lettere for den, der vil skrive en professionel applikation.

Det fjerde og sidste bibliotek, der er med i pakken, hedder SCOM, og det har UniComal konstrueret i samarbejde med Dansk System Elektronik. Med dette er det muligt at kommunikere med ydre enheder gennem den serielle RS-232C-port.

Det er måske nok på det industrielle område, at UniComal har sine største muligheder. Netop ved styring og regulering af maskiner har man glæde af et interaktivt programmeringssprog. Under opbygning kan man give kommandoerne til systemet, og resultatet vil blive synligt direkte på den tilkoblede enhed. Når man så har sikret, at styringsprogrammet er korrekt, kan man forlade fortolkeren og foretage en oversættelse, hvorved det færdige produkt fremstår.

Det er da næppe heller en tilfældighed, at UniComal er gået i samarbejde med adskillige danske industrivirksomheder. I Dansk System Elektronik benyttes UniComal blandt andet til dataopsamling og processtyring, og som et andet eksempel kan nævnes, at lufthavnen i Billund benytter et UniComal-program til behandling af meteorologiske data. UniComal er således med til at sikre, at fly kan komme korrekt op

og ned i tåget vejr.

En primitiv oversætter

Blandt de mange eksempler og moduler på disketterne findes som nævnt også en oversætter. Dens opgave er at oversætte syntaktisk korrekt kode til en eksekverbar fil.

Det første møde med denne var temmeligt frustrerende. Som tidligere skrevet var manualen til oversætteren ikke færdig, da vi lånte et eksemplar til test, og vi måtte nøjes med en midlertidig, elektronisk manual på 1.500 skærmlinjer.

Oversætteren er baseret på kommandolinjeprincippet, hvorfor der ved opstart spørges efter den fil, hvorpå kildeteksten befinder sig. Derefter skal man svare på en række spørgsmål såsom inkludering af eksterne moduler, placering af den eksekverbare fil, stakstørrelse, programstørrelse og ganske bemærkelsesværdigt, hvilken ledsagende tekst man ønsker forbundet med den endelige fil. Denne vil blive indlagt, og man kan altid ved en simpel kommando spørge på indholdet af denne tekst. Der kan også lægges angivelse af serienummer og lignende sager ind, hvorfor en direkte nummerering af færdige eksemplarer af en applikation er mulig.

Man må sige, at det i vore dage – med tanke på integrerede udviklingsmiljøer – er en besværlig fremgangsmåde til oversættelse, men det virker da.

I stedet for at besvare spørgsmålene undervejs kan man angive parametre i kommandolinjen ved kaldet af oversætteren, og her kan også angives oversætterdirektiver såsom brug af matematikprocessor (eller ej) og et valg mellem direkte skrivning i skærmmukommelsen eller kun over BIOS.

Der er desværre en tendens til, at man drukner i underlige parameterkonventioner og skråstreg, og vi fandt ikke denne måde at

UniComal


```

0010 // Program til afprøvning af modul indeholdende Bubble Sort-sortering.
0020 // (C) Alt om Data
0030 //
0040 // Modulen indeholdende en ombytningsprocedure
0050 //
0060 MODULE basis
0070 PROC ombyt(REF a, REF b)
0080   temp:=a
0090   a:=b
0100   b:=temp
0110 ENDPROC ombyt
0120 ENDMODULE basis
0130 //
0140 // Modulen bubble indeholdende sorteringsproceduren bubblesort
0150 //
0160 MODULE bubble
0170   USE basis
0180   PROC bubblesort(REF tabel(), maxindeks)
0190     TRAP
0200       IF maxindeks>20 THEN
0210         REPORT
0220       ENDIF
0230       i:=1
0240       ombytning:=TRUE
0250       WHILE (i<maxindeks-1) AND (ombytning) DO
0260         j:=1
0270         ombytning:=FALSE
0280         WHILE (j<maxindeks-i) DO
0290           IF tabel(j)>tabel(j+1) THEN
0300             ombyt(tabel(j), tabel(j+1))
0310             ombytning:=TRUE
0320           ENDIF
0330           j:=j+1
0340         ENDWHILE
0350         i:=i+1
0360       ENDWHILE
0370       HANDLER // fejlbehandling ved for højt indeks
0380         PRINT "Der er angivet et for højt maksimalt indeks."
0390       ENDTRAP
0400     ENDPROC bubblesort
0410   ENDMODULE bubble
0420 //
0430 // Test af bubblesort-proceduren
0440 //
0450 USE bubble
0460 PRINT
0470 PRINT "Start program."
0480 PRINT
0490 DIM t(1:20)
0500 t(1):=33
0510 t(2):=77
0520 t(3):=21
0530 t(4):=80
0540 t(5):=14
0550 bubblesort(t(), 5)
0560 FOR i:=1 TO 5 DO
0570   PRINT t(i), TAB(5)
0580 ENDFOR i
0590 //
0600 // Test af fejlsituation
0610 //
0620 PRINT
0630 bubblesort(t(), 40)
0640 PRINT
0650 PRINT "Stop program."
0660 END

```

Et program skrevet under UniComal til kontrol af en implementation af sorteringsmetoden Bubble Sort. Bemærk brugen af moduler og fejlbehandlingsfaciliteter.

benytte en oversætter særlig behagelig. Brugervenligheden er i bund, og hvis du sidestiller arbejdsmetoden med den, der bruges i DOS fra før 4.0-dagene, så har du et meget godt billede af dømmet.

Da vi havde skrevet det første

UniComal-program, havde forladt fortolkeren og forsøgte en oversættelse, så blev det nægtet os på det bestemteste. Programmet var nemlig endnu ikke blevet skannet i fortolkeren. Dette er en forudsætning for, at det kan oversættes, også selvom det er fejl-

frit.

Endvidere var det en kilde til irritation, at hvis der under angivelsen af parametre i oversættelsesprocessen blev skrevet den mindste fejl, så blev oversætteren forladt prompte. Der var kun den mulighed at starte forfra. Man lærer på den måde hurtigt at skrive korrekte svar på forespørgslerne.

Vi fik dog skrevet programmer alligevel, og det var interessant at se, hvor meget de fyldte. Et program kun indeholdende én linje, hvorpå der var givet en tom kommentar, fyldte i eksekverbar form godt 60 Kb. Hvilken information, der er gemt i disse, er ikke godt at vide.

Sæt din egen konfiguration

Ikke sjældent er man utilfreds med den konfiguration, som et produkt leveres med, og her åbner UniComal mulighed for opbygning af en særlig fil til ens egen foretrukne konfiguration. Denne skal så indeholde de oplysninger om skærm (farver og markørudseende), hukommelsesstørrelse (stak, program etc.), tastaturopsætning (fx funktionstaster), stier til søgning efter eksterne moduler i fil og anden information af samme skuffe.

Med konstruktion af denne fil bliver standardopsætningen overskrevet på de punkter, hvor det er angivet.

Tankegangen er på dette punkt fuldstændig lig de øvrige steder i UniComal. Du må selv gå ind og skrive denne fil og kontrollere i manualen, at du har husket alt. Der er ingen forslag til, hvad du kan have med, og der er ingen menu, der dumper ned i hovedet på dig, hvor du vælger ud fra angivne muligheder. Vi fandt det svært at vænne os til denne arbejdsmetode, som virker gammeldags i forhold til moderne systemer. Selvom den kan forsvares til at være fuldt tilstrækkelig for den kyndige bruger, så kan brugervenligheden i hvert fald kon-

kluderes som værende mindre god. Dertil kræver opbygningen af en sådan fil for meget indsigt i forhold, der, efter vor mening, burde være brugeren uvedkommende.

Fleksibilitet

UniComal har vi testet i en PC-udgave kørende under DOS. Der findes også en OS/2-udgave, og UniComal fortæller, at der er en UNIX-udgave under udvikling. Den grundlæggende ide bag hele dette projekt er, at et hvilket som helst UniComal-program kan flyttes fra at køre under det ene operativsystem til at køre under det andet.

Dette gør UniComal særdeles stærkt, da en programmør ikke behøver tænke nærmere på de ydre, fysiske rammer, når blot grænsefladen i form af procedurer og funktioner i diverse moduler er kendt.

Konklusion

UniComal version 3.00 er en stærkt forbedret udgave af Comal-sproget udbygget med en lang række forskellige biblioteker til alskens forskellige opgaver. Hvis man derfor finder sproget betagende og har behov for en interaktiv dialog ved udviklingsprocessen, så er det en god pakke.

Dette er særligt relevant ved benyttelse i industrien, hvor modulet til kommunikation over den serielle port må fremhæves.

Selve arbejdsgangen i UniComal er tung. Til trods for, at editoren er syntaksstyret og skærmorienteret, vænnede vi os aldrig rigtig til den – måske på grund af indgroede vaner.

Der er ingen hjælp at hente online, men til gengæld må manualerne alt i alt kategoriseres som brugbare med henblik på opslag. En Comal-novice eller en PC-ukyndig får ingen assistance gennem den medfølgende dokumentation. □

Check Out:

UniComal version 3.00 forhandles af UniComal A/S, Søborg, tlf. 31 67 35 11. Prisen er kr. 5.250 excl. moms med tillæg af kr. 750 excl. moms i OS/2-udgave.